

# Мечи залива работорговцев

Работа выполнена студентами 312 гр:

- Арбузов П.А.
- Кюнченкова Д.Д.
- Семенов А.В.

## Постановка задачи

После оглушительного успеха в освобождении Астапора, Миэрина и Юнкя от власти работорговцев Дейнерис Бурерожденная открыла себе доступ к Летнему морю, а следовательно — путь в Вестерос. Для ведения войны с Семью Королевствами нужно оружие, а для оружия нужна сталь. Нет никаких сомнений в кузнечном искусстве Безупречных, однако поставщики стали не столь надежны. Два основных поставщика стали — это Westeros Inc. и Harpy & Co. На протяжении нескольких месяцев мы закупаем сталь у обеих компаний, и каждая из них предлагает ощутимую скидку при заключении эксклюзивного договора на поставку. Советник королевы Тирион Ланнистер знает о твоём умении принимать взвешенные рациональные решения и просит помощи в объективном решении вопроса о том, с какой из компаний следует заключить эксклюзивный договор на поставку стали. У Тириона есть записи о производстве мечей каждым из кузнецов-безупречных, а также данные о количестве сломанных мечей в каждый из месяцев ведения боевых действий.

## Цель работы

Необходимо провести разведовательный анализ данных с целью ответа на вопрос "С каким из поставщиков стали следует заключить договор?"

# Исходные данные

Дан CSV-файл с данными о производстве оружия и количестве единиц сломанного оружия за каждый месяц каждым из кузнецов. В файле даны данные о двух компаниях поставщиках стали. Все кузнецы производят оружие одинаково, так что качество оружия зависит лишь от поставщика стали. Исходный файл:

- В колонке "unsullen.id" задан номер кузнеца, проводившего работу.
- В колонке "production.date" задан месяц производства
- В колонке "report.date" задан месяц отчета о производстве
- В колонке "produced" задано сколько мечей было произведено в соответствующий месяц
- В колонке "defects" задано сколько мечей было сломано в месяц отчета, произведенных в соответствующий месяц производства
- В колонке "supplier" указан соответствующий поставщик стали

## Решение

*Разведовательный анализ данных - это подход к анализу данных для выявления их основных характеристик, часто с помощью методов визуализации данных*

### Загрузка данных

Считываем данные из CSV-файла в удобном для работы формате с помощью функции `pandas.read_csv(file, sep)` из модуля **pandas**. Далее, с помощью метода `loc` разделяем данные по производителям стали.

## Анализ данных и построение графиков

С помощью методов *groupby* и *sum* а так же функций из модуля **matplotlib.pyplot** строим гистограмму с показателями по сумме количества произведенного оружия из стали каждого из поставщиков.

Далее, с помощью метода *hist* строим гистограмму частоты количества сломаного оружия для каждого из поставщиков стали.

Далее, мы считаем среднюю статистику сломанных мечей для каждого производителя применяя методы и функции из модулей **pandas** и **numpy**. Поскольку в нашем ряду распределения значения количества сломанных мечей встречаются не с одинаковой частотой, то мы считаем *взвешенное среднее*:

$$\bar{x} = \frac{\sum x_i f_i}{\sum f_i}$$

где  $x_i$  - это значения, а  $f_i$  - частота встречаемости этих значений.

Полученные результаты отражаем в виде графика **Box Plot**. На диаграмме размаха отражены верхние и нижние границы, квартили, среднее значение, а так же выбросы. Наш график отражает какое количество мечей в среднем ломается на какой месяц после изготовления.

Затем мы так же строим графики, отражающие среднее количество сломанных мечей на каждый месяц после изготовления. Из последних графиков уже можно сделать вывод о результатах нашего анализа.

## Используемые средства

Модули:

- pandas
- numpy
- matplotlib.pyplot

## Функции:

- `pandas.read_csv(file, sep)` - считывает CSV-файл с заданными разделителями в `DataFrame`
- `pandas.DataFrame.loc(label)` - возвращает часть данных из `DataFrame`, например по заданной метке `label`
- `pandas.DataFrame.hist()` - создает гистограмму данных из данного `DataFrame`
- `pandas.Series.unique()` - возвращает уникальные значения из `Series`
- `pandas.DataFrame.mean()` - возвращает среднее значение от переданных данных
- `numpy.multiply(arr1, arr2)` - перемножает два массива, возвращает массив `numpy`.
- `matplotlib.pyplot.boxplot` - строит `BoxPlot` на переданных данных